

Research Article

A Unified Framework for DPLL(T) + Certificates

Min Zhou,^{1,2,3,4} Fei He,^{1,2,3} Bow-Yaw Wang,⁵ Ming Gu,^{1,2,3} and Jianguang Sun^{1,2,3}

Tsinghua National Laboratory for Information Science and Technology (TNList), Beijing , China

School of Software, Tsinghua University, Beijing , China

Key Laboratory for Information System Security, MOE, Beijing , China

Department of Computer Science and Technologies, Tsinghua University, Beijing , China

Institute of Information Science, Academia Sinica, Taipei , Taiwan

Correspondence should be addressed to Min Zhou; zhoumin@gmail.com

Received February ; Accepted April

Academic Editor: Xiaoyu Song

Copyright © 2014 Min Zhou et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Satisfiability Modulo theories (SMT) techniques are widely used nowadays. SMT solvers are typically used as verification backends. When an SMT solver is invoked, it is quite important to ensure the correctness of its results. To address this problem, we propose a unified certificate framework based on DPLL(T), including a uniform certificate format, a unified certificate generation procedure, and a unified certificate checking procedure. The certificate format is shown to be simple, clean, and extensible to different background theories. The certificate generation procedure is well adapted to most DPLL(T)-based SMT solvers. The soundness and completeness for DPLL(T) + certificates were established. The certificate checking procedure is straightforward and efficient. Experimental results show that the overhead for certificate generation is only 10%, which outperforms other methods, and the certificate checking procedure is quite time saving.

1. Introduction

1.1. Background and Motivation. Satisfiability Modulo theories (SMT) techniques are getting increasingly popular in many verification applications. An SMT solver takes as input an arbitrary formula given in a specific fragment of first order logic with interpreted symbols. It returns a judgment telling whether the formula is satisfiable. By satisfiable, it means that there exists at least one interpretation under which the formula evaluates to true.

SMT solvers are typically used as verification backends. During a verification process, SMT solver may be invoked many times. Each time the SMT solver is given a complex logical formula and is expected to give a correct judgment. However, SMT solvers employ many sophisticated data structures and tricky algorithms, which make their implementations prone to error. For instance, there are some solvers disqualified in SMT-COMP due to their incorrect judgments [1]. We therefore strongly believe that SMT solvers should be supported with a formal assessment of their correctness.

Ensuring the correctness of an SMT solver is never easy. Generally, there are two approaches: to verify the SMT solver or to certify their judgments. For the former approach, it proves directly that the solver is correct. So, its judgments are naturally correct. Though rigor in logic, this method is not generally practical because verifying an SMT solver requires great workload. Not only the algorithm but also the implementation should be verified in a formal way. To the best of our knowledge, no state-of-the-art SMT solver has been completely verified. More importantly, this approach is solver dependent. Even if a solver is completely verified, the whole verification has to be done again when the solver is modified.

The rationale of the latter approach is as follows: instead of proving the correctness of the solver, we prove its judgment. For this purpose, the solver is required to return a piece of evidence to support its judgment. This evidence is called *certificate* in this paper. Then, the correctness for the judgment is ensured by the certificate. If so, a rigorous and detailed proof can be constructed by referring to the certificate. Obviously, checking a certificate is much easier than

